# SHADERS AND SHADING

Template 2

Computer Graphics 2

# What's New?

- Ray carries hit normal
- Light
- Shaders

# Hit Normal

- Normal of objects' surface at intersection point of a ray with an object
  - How to calculate it for plane and sphere?
- Used in calculation of illumination

# Light

- Various types of light sources
  - Directional light, spot light, point light, area light
- Each light has
  - Intensity – defines strength with which light illuminates the scene
  - Color – defines the color of the light
    - Diffuse color
    - Specular color
    - Ambient color

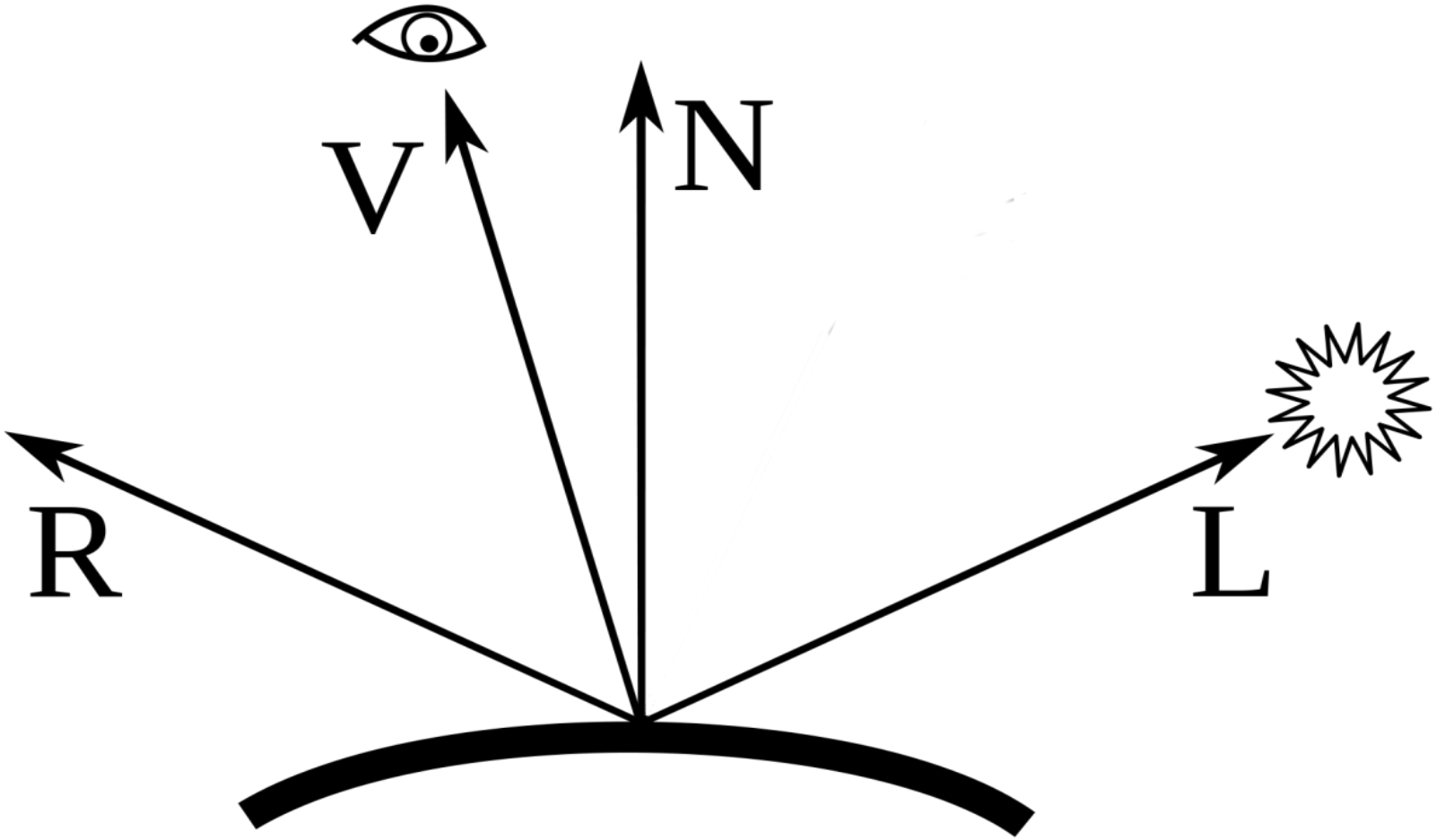
# Shader

- Used to define color at a point
- Color is usually calculated using:
  - Point in the scene
  - Normal of points' surface
  - Direction from point to eye
  - Direction from point to light source
  - Light intensity and color at point

# Phong Shader

- Local illumination model
- Not physically based, does not support:
  - Helmholtz reciprocity
  - Conserving energy
- Split light into components:
  - Ambient – constant for the material
  - Diffuse – depends on position of the light
  - Specular – depends on light and eye position

# Phong Shader - Illustration



# Phong Ambient

$$I_{ambient} = k_a I_a$$

- Simulates light incoming from objects in the scene
- No physical basis – just a constant
- $k_a$ object ambient constant
- $I_a$ ambient light color of a light source

# Phong Diffuse

$$I_{diff} = k_d I_d (\boldsymbol{l} \cdot \boldsymbol{n})$$

- Lambertian diffuse reflection
- $k_d$ object diffuse constant
- $I_d$ incoming light diffuse color
  - Scaled by light intensity
- $(\boldsymbol{l} \cdot \boldsymbol{n})$ angle between illuminated point normal and incoming light direction

# Phong Specular

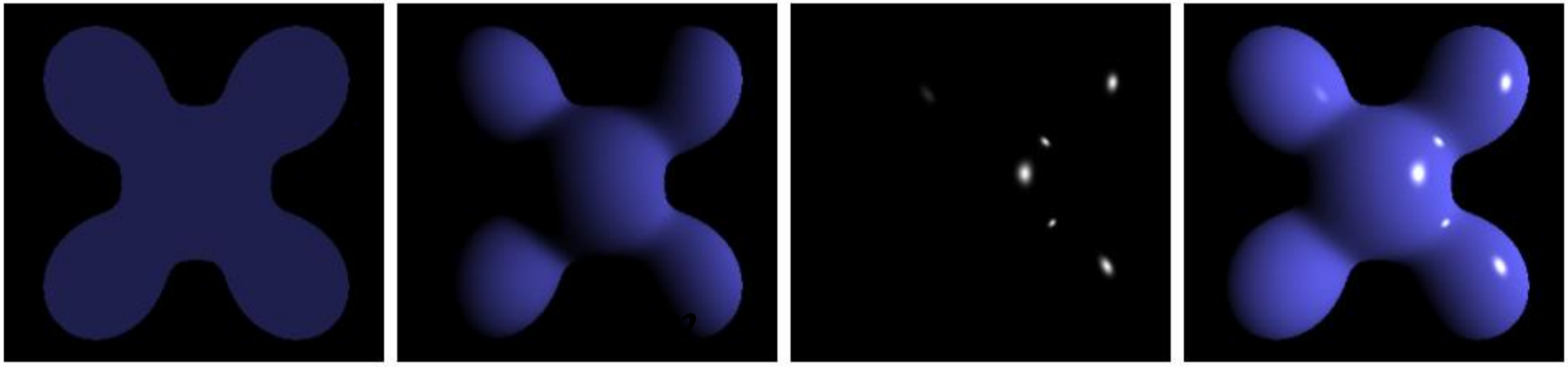
$$I_{spec} = k_s I_l (\boldsymbol{r} \cdot \boldsymbol{v})^{n_s}$$

- Specular reflection in direction of perfect glossy reflection
- $k_s$ object specular constant
- $I_l$ incoming light specular color
  - Scaled by light intensity
- $\boldsymbol{r}$ light vector reflected along point normal
- $\boldsymbol{v}$ view direction
- $(\boldsymbol{r} \cdot \boldsymbol{v})$ angle between view direction and reflected vector
- $n_s$ shinines

# Phong Shader – Putting It All Together

$$I = I_{ambient} + I_{diff} + I_{spec} = k_a I_a + k_d I_d(\boldsymbol{l} \cdot \boldsymbol{n}) + k_s I_s(\boldsymbol{r} \cdot \boldsymbol{v})^{n_s}$$

$$I = k_a I_a + \sum_{i=1}^{n} ( k_d I_{i,d}(\boldsymbol{l_i} \cdot \boldsymbol{n}) + k_s I_{i,s}(\boldsymbol{r_i} \cdot \boldsymbol{v})^{n_s})$$



# Questions?